# Principles of Robot Autonomy I

Introduction to localization and filtering theory

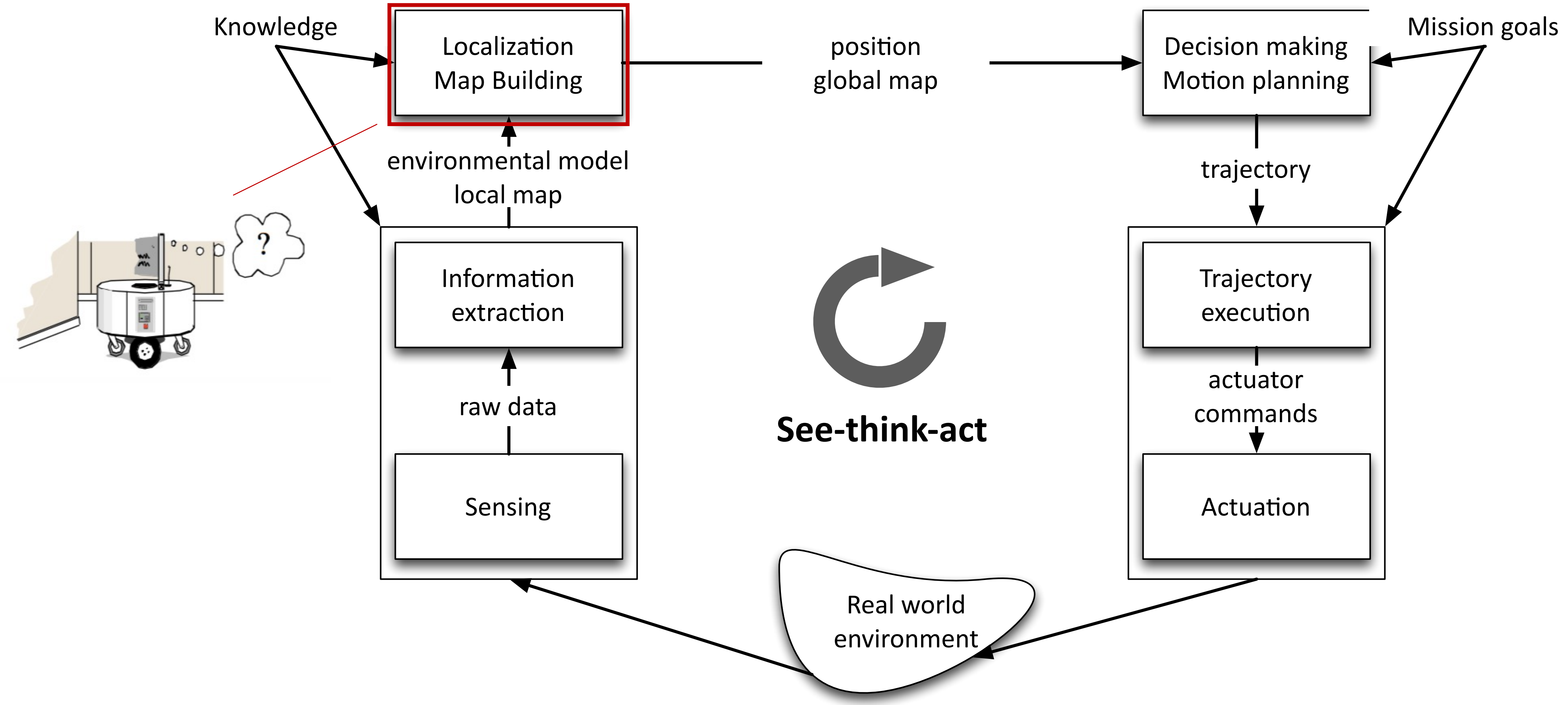
Stanford University

Autonomous Systems Lab

# Logistics

- Additional section slot: Friday 9:45—11:45AM
- Genbu down for PY3 upgrade this Wednesday (10/27)
    - Be sure to preserve your work (e.g., using git)!
    - HW3, section 5 deadlines both extended +1 day

# Module 3


Knowledge
Localization
Map Building
position
global map
Decision making
Motion planning
Mission goals
environmental model
local map
trajectory
Information
extraction
Trajectory
execution
raw data
See-think-act
actuator
commands
Sensing
Actuation
Real world
environment

# Today’s lecture

- Aim
  - Learn basic concepts about Bayesian filtering

- Readings
  - S. Thrun, W. Burgard, and D. Fox. Probabilistic robotics. MIT press, 2005. Chapter 2

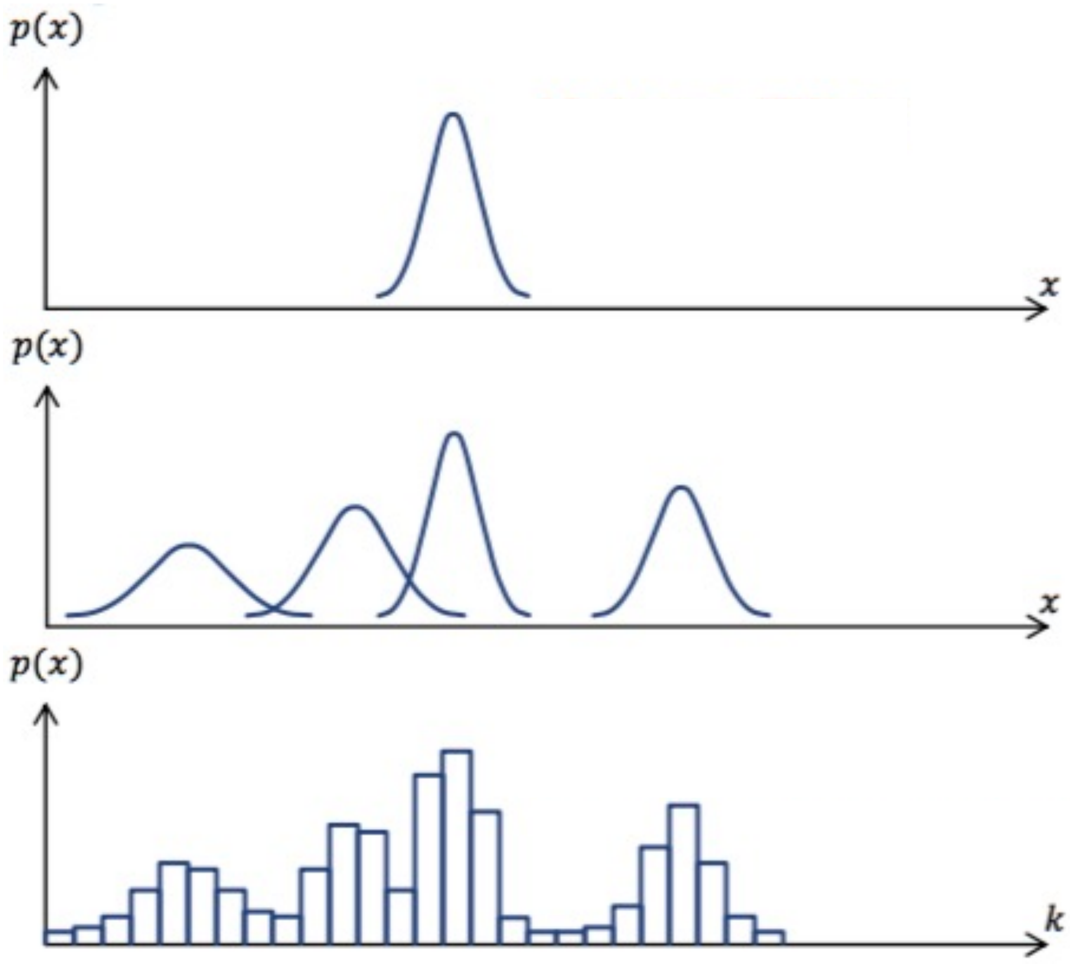
# Localization

- Two main approaches:
  1. Behavioral approach: design a set of behaviors that together result in the desired robot motion (no need for a map)
  2. Map-based approaches: robot *explicitly* attempts to localize by collecting sensor data, then updating belief about its position with respect to a map

- We will focus on map-based approaches; two main aspects:
  - Map representation: how to represent the environment?
  - Belief representation: how to model the belief regarding the position within the map?

# *Probabilistic* map-based localization

- Key idea: represent belief as a probability distribution
    1. Encodes sense of position
    2. Maintains notion of robot's uncertainty
- Belief representation:
    1. Single-hypothesis vs. multiple hypothesis
    2. Continuous vs. discretized
- Today we will overview basic concepts in Bayesian filtering



# Basic concepts in probability

- Key idea: quantities such as sensor measurements, states of a robot, and its environment are modeled as random variables (RVs)
- Discrete RV: the space of all the values that a random variable *X* can take on is *discrete*; characterized by probability mass function (pmf)

$$\underbrace{p(X}_{\text{Random variable}} = \underbrace{x)}_{\text{Specific value}} \quad (\text{or } p(x)), \qquad \sum_x p(X = x) = 1$$

- Continuous RV: the space of all the values that a random variable *X* can take on is *continuous;* characterized by probability density function (pdf)

$$P(a \leq X \leq b) = \int_a^b p(x)\, dx, \qquad \int_{-\infty}^{\infty} p(x)\, dx = 1$$

# Joint distribution, independence, and conditioning

- Joint distribution of two random variables *X* and *Y* is denoted as

$$p(x, y) := p(X = x \text{ and } Y = y)$$

- If *X* and *Y* are independent

$$p(x, y) = p(x)p(y)$$

- Suppose we know that $Y = y$ (with $p(y) > 0$); conditioned on this fact, the probability that the $X$'s value is $x$ is given by

$$p(x \mid y) := \frac{p(x, y)}{p(y)}$$

Conditional probability

Note: if *X* and *Y* are independent

$$p(x \mid y) := p(x)!$$

# Law of total probability

- For discrete RVs:

$$p(x) = \sum_y p(x, y) = \sum_y p(x \mid y) p(y)$$

- For continuous RVs:

$$p(x) = \int p(x, y) dy = \int p(x \mid y) p(y) dy$$

- Note: if $p(y) = 0$, define the product $p(x \mid y)p(y) = 0$

# Bayes' rule

- Key relation between $p(x \mid y)$ and its "inverse," $p(y \mid x)$
- For discrete RVs:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\sum_{x'} p(y \mid x')p(x')}$$

- For continuous RVs:

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)} = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\, dx'}$$

# Bayes' rule and probabilistic inference

- Assume *x* is a quantity we would like to infer from *y*
- Bayes rule allows us to do so through the inverse probability, which specifies the probability of data *y* assuming that *x* was the cause

Posterior probability distribution

Prior probability distribution

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int p(y \mid x')p(x')\, dx'}$$

Data

Normalizer, does not depend on $x := \eta^{-1}$

- Notational simplification

$$p(x \mid y) = \eta\, p(y \mid x)p(x)$$

# More on Bayes' rule and independence

- Extension of Bayes rule: conditioning Bayes rule on *Z=z* gives

$$p(x \mid y, z) = \frac{p(y \mid x, z)p(x \mid z)}{p(y \mid z)}$$

- Extension of independence: *conditional independence*

$$p(x, y \mid z) = p(x \mid z)p(y \mid z), \qquad \text{equivalent to} \begin{cases} p(x \mid z) = p(x \mid z, y) \\ p(y \mid z) = p(y \mid z, x) \end{cases}$$

- Note: in general

$$p(x, y \mid z) = p(x \mid z)p(y \mid z) \not\Rightarrow p(x, y) = p(x)p(y)$$

$$p(x, y) = p(x)p(y) \not\Rightarrow p(x, y \mid z) = p(x \mid z)p(y \mid z)$$

# Expectation of a RV

- Expectation for discrete RVs: $E[X] = \sum_x x\, p(x)$
- Expectation for continuous RVs: $E[X] = \int x\, p(x)\, dx$
- Expectation is a linear operator: $E[aX + b] = a\, E[X] + b$
- Expectation of a vector of RVs is simply the vector of expectations
- Covariance

$$\mathrm{cov}(X, Y) = E[(X - E[X])(Y - E[Y])^T] = E[XY^T] - E[X]E[Y]^T$$

# Model for robot-environment interaction

- Two fundamental types of robot-environment interactions: the robot can influence the state of its environment through control actions, and gather information about the state through measurements
- State $x_t$: collection at time *t* of all aspects of the robot and its environment that can impact the future
  - Robot pose (e.g., robot location and orientation)
  - Robot velocity
  - Locations and features of surrounding objects in the environment, etc.
- Useful notation: $x_{t_1:t_2} := x_{t_1}, x_{t_1+1}, x_{t_1+2}, \ldots, x_{t_2}$
- A state $x_t$ is called *complete* if no variables prior to $x_t$ can influence the evolution of future states → Markov property

# Measurement and control data

- Measurement data $z_t$: information about state of the environment at time $t$; useful notation

$$z_{t_1:t_2} := z_{t_1}, z_{t_1+1}, z_{t_1+2}, \ldots, z_{t_2}$$

- Control data $u_t$: information about the change of state at time $t$; useful notation

$$u_{t_1:t_2} := u_{t_1}, u_{t_1+1}, u_{t_1+2}, \ldots, u_{t_2}$$

- Key difference: measurement data tends to increase robot's knowledge, while control actions tend to induce a loss of knowledge

# State equation

- General probabilistic generative model

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t})$$

Convention: first take control action and then take measurement

- Key assumption: state is complete (i.e., the Markov property holds)

$$p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t \mid x_{t-1}, u_t)$$

State transition probability

- In other words, we assume *conditional independence*, with respect to conditioning on $x_{t-1}$

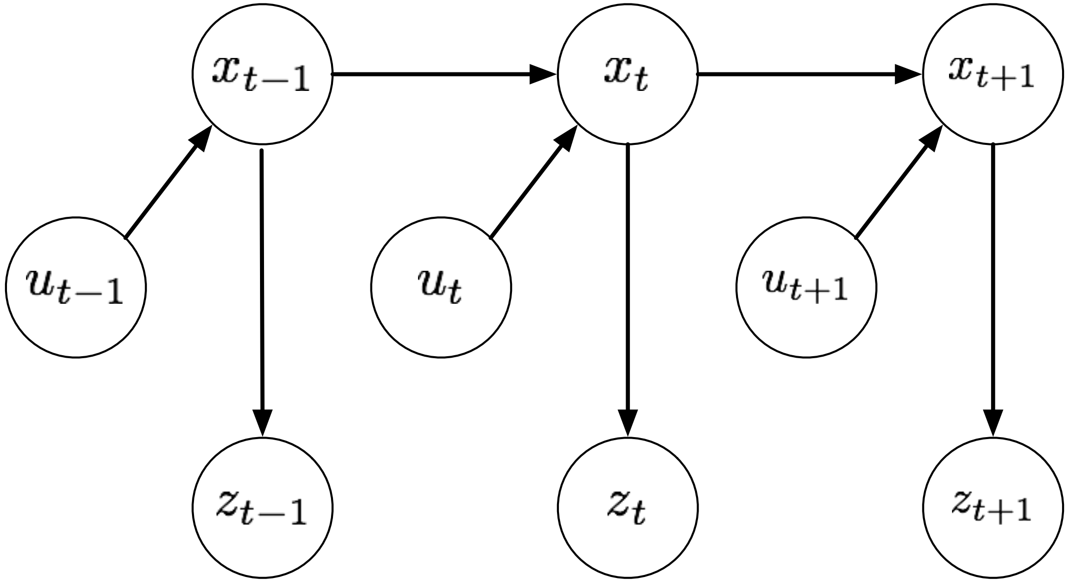
# Measurement equation and overall stochastic model

- Assuming $x_t$ is complete

$$p(z_t \,|\, x_{0:t},\, z_{1:t-1},\, u_{1:t}) = p(z_t \,|\, x_t)$$

Measurement probability

- Overall dynamic Bayes network model (also referred to as hidden Markov model)



# Belief distribution

- Belief distribution: reflects internal knowledge about the state
- A belief distribution assigns a probability to each possible hypothesis with regard to the true state
- Formally, belief distributions are posterior probabilities over state variables conditioned on the available data

$$bel(x_t) := p(x_t \mid z_{1:t}, u_{1:t})$$

- Similarly, the *prediction* distribution is defined as

$$\overline{bel}(x_t) := p(x_t \mid z_{1:t-1}, u_{1:t})$$

- Calculating $bel(x_t)$ from $\overline{bel}\ (x_t)$ is called correction or measurement update

# Bayes filter algorithm

- Bayes' filter algorithm: most general algorithm for calculating beliefs
- Key assumption: state is complete

Update rule

- Recursive algorithm
  - Step 1 (prediction): compute $\overline{bel}\,(x_t)$
  - Step 2 (measurement update): compute $bel(x_t)$
- Algorithm initialized with $bel(x_0)$ (e.g., uniform or points mass)

**Data:** $bel(x_{t-1}), u_t, z_t$
**Result:** $bel(x_t)$
**foreach** $x_t$ **do**
$\quad \overline{bel}(x_t) = \int p(x_t \mid u_t, x_{t-1})\, bel(x_{t-1})\, dx_{t-1};$
$\quad bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t);$
**end**
Return $bel(x_t)$

# Derivation: measurement update

$$
\begin{aligned}
bel(x_t) &= p(x_t \mid z_{1:t},\, u_{1:t}) \\
&= \frac{p(z_t \mid x_t, z_{1:t-1},\, u_{1:t})\, p(x_t \mid z_{1:t-1},\, u_{1:t})}{\underbrace{p(z_t \mid z_{1:t-1},\, u_{1:t})}_{:=\eta^{-1}}} && \text{Bayes rule} \\
&= \eta\, p(z_t \mid x_t)\ \underbrace{p(x_t \mid z_{1:t-1},\, u_{1:t})}_{=\overline{bel}(x_t)} && \text{Markov property}
\end{aligned}
$$

# Derivation: correction update

$$\overline{bel}(x_t) = p(x_t \,|\, z_{1:t-1},\, u_{1:t})$$

$$= \int p(x_t \,|\, x_{t-1},\, z_{1:t-1},\, u_{1:t})\, p(x_{t-1} \,|\, z_{1:t-1},\, u_{1:t})\, dx_{t-1}$$

Total probability

$$= \int p(x_t \,|\, x_{t-1},\, u_t)\, p(x_{t-1} \,|\, z_{1:t-1},\, u_{1:t})\, dx_{t-1}$$

Markov

$$= \int p(x_t \,|\, x_{t-1},\, u_t)\, p(x_{t-1} \,|\, z_{1:t-1},\, u_{1:t-1})\, dx_{t-1}$$

For general output feedback policies, $u_t$ does not provide additional information on $x_{t-1}$

$$= \int p(x_t \,|\, x_{t-1},\, u_t)\, bel(x_{t-1})\, dx_{t-1}$$

# Discrete Bayes' filter

- Discrete Bayes' filter algorithm: applies to problems with *finite* state spaces

- Belief $bel(x_t)$ represented as pmf $\{p_{k,t}\}$

**Data:** $\{p_{k,t-1}\}, u_t, z_t$
**Result:** $\{p_{k,t}\}$
**foreach** $k$ **do**
$\quad \overline{p}_{k,t} = \sum_i p(X_t = x_k \,|\, u_t, X_{t-1} = x_i)\, p_{i,t-1};$
$\quad p_{k,t} = \eta\, p(z_t \,|\, X_t = x_k)\, \overline{p}_{k,t};$
**end**
Return $\{p_{k,t}\}$

# Next time